

Node.js Project Ideas

1. File Management System (2-3 days)

Description: Build a CLI-based file management tool to handle basic file system operations.

Features:

- Create, read, update, and delete files.
- Rename files and directories.
- Append content to existing files.
- List all files and subdirectories in a given directory.
- Check if a file or directory exists.

2. HTTP Server with Custom Routing (2 days)

Description: Create an HTTP server that handles multiple routes dynamically.

Features:

- /: Serve a homepage with basic HTML.
- /about: Serve an "About Us" page.
- /contact: Serve a contact form.
- /404: Handle undefined routes with a custom error page.

3. Basic REST API (3-4 days)

Description: Build a REST API to manage a collection of items (e.g., books, tasks).

Features:

- GET /items: Fetch all items.
- POST /items: Add a new item.
- PUT /items/:id: Update an item by ID.

- DELETE /items/:id: Delete an item by ID.
- Store the data in a local file (e.g., items.json).

4. Command-Line To-Do App (2 days)

Description: Create a CLI-based to-do list application.

Features:

- Add tasks to the list.
- Mark tasks as complete or incomplete.
- Delete tasks.
- View the current list of tasks.

5. Static File Server (2-3 days)

Description: Create a server that serves static files (e.g., HTML, CSS, JS).

Features:

- Serve files from a public directory.
- Handle 404 for non-existent files.
- Log requests to a file (e.g., access.log).

6. Simple Chat Application (4-5 days)

Description: Create a real-time chat application using Node.js.

Features:

- A basic terminal-based chat system for multiple users.
- Users can join or leave the chat.
- Messages are broadcasted to all connected users.

7. URL Shortener (3-4 days)

Description: Build a URL shortening service similar to TinyURL.

Features:

- Accept long URLs and generate short URLs.
- Store URL mappings in a file.
- Redirect short URLs to the original URL.
- Basic analytics (e.g., how many times each URL was accessed).

8. Markdown to HTML Converter (2-3 days)

Description: Create a CLI tool to convert Markdown files into HTML.

Features:

- Input: A .md file.
- Output: An equivalent .html file.
- Handle basic Markdown syntax (e.g., headings, paragraphs, links).

9. Weather App (4-5 days)

Description: Create a command-line weather application.

Features:

- Fetch current weather data for a given city using an API (e.g., OpenWeather).
- Display temperature, humidity, and weather description.
- Cache recent weather data in a file for quick access.

10. Basic Authentication System (5-7 days)

Description: Build a simple system to handle user registration and login.

Features:

- Users can register with a username and password.
- Passwords are hashed before storage.
- Users can log in and receive a success or failure response.
- Store user data in a local file.

11. Log File Analyzer (3-4 days)

Description: Analyze a log file (e.g., server logs) to extract insights.

Features:

- Count the number of requests.
- Find the most accessed endpoint.
- Display the number of 404 errors.